Email Spam Classifier

BY

MRS. SWATI AMIT MOTUGADE

INTRODUCTION

Spam emails can be not only annoying but also dangerous to consumers.

Spam emails can be defined as:

1. Anonymity
2. Mass Mailings
3. Unsolicited

The spam emails are the messages randomly sent to multiple addresses by all sort of groups, but mostly lazy advertisers and criminals who wish to lead you to phishing sites.

Naive-Bayes Classifier

It is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combination of values in a given dataset.

It represents as a vector of feature values.

It is very useful in proper classification of emails.

The precision of this method is known to be very effective.

Problem Statement

A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available.

The main goal of this project is to build a spam filtering system from scratch.

Objective

The main objectives of identification of spam emails are:

1. To give knowledge to the user about the fake emails and relevant emails.
2. To classify that a particular mail is spam or not.

Literature Review

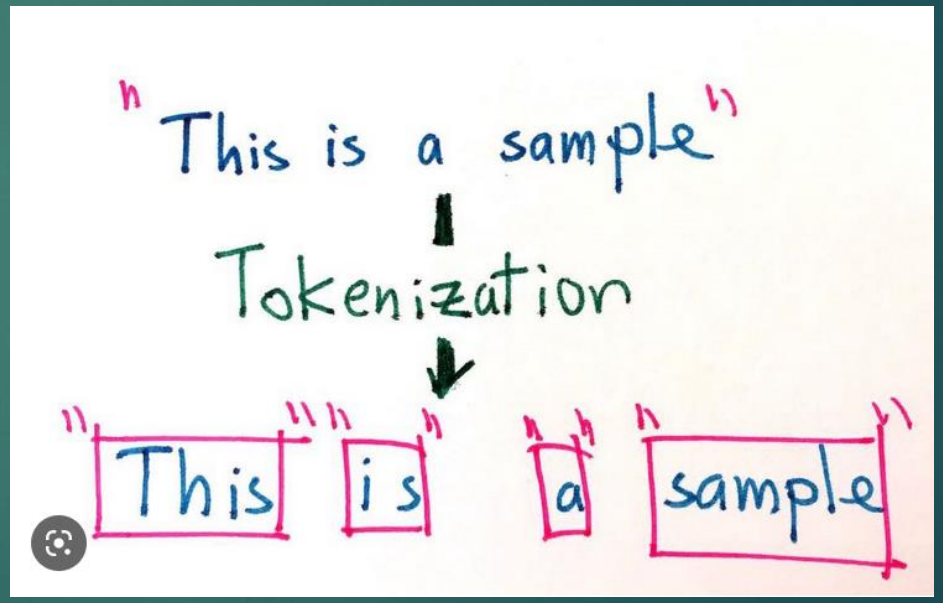
- Consulted from G. He, Spam Detection, 1st ed. 2007 and learned about this problem.
- Spam prevention is often neglected, although some simple measures can dramatically reduce the amount of spam that reaches your mailbox.
- Before they are able to send you spam, spammers obviously first need to obtain your email address, which they can do through different routes.

Data Preprocessing

Tokenization:

Tokenization is the process of breaking a stream of text up into words, phrases, symbols or other meaningful elements called tokens.

The list of tokens becomes input for further processing such as parsing or text mining.



Lemmatization

- Lemmatization in linguistics, is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.
- In computational linguistics, lemmatization is the algorithmic process of determining the lemma for a given word.

Removal of Stop words

Sometimes, the extremely common words which would appear to be of very little value in helping select documents matching user need are excluded from the vocabulary entirely.

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Requirement Analysis

- Functional Requirement:

  To classify the emails which is done by first taking out the feature vector extraction which involves first taking out whether the word is a spam or not.

- Non Functional Requirement:

  Ensures high availability of email data or dataset

  User should get the result as fast as possible.

  It should be easy to use i.e., user is just required to type the words and click then the result is displayed or user is just required to enter a pair of reasonable sentence.

Testing

- We tested the dataset and found out which email is a spam and which is not spam or in other words ‘ham’ . Spam emails are labelled as 1 and ham mails are labelled as 0.
- We calculated the feature vector to know whether it is spam or ham.
- Using this feature vector Naive-Bayes Algorithm works by comparing the training dataset to the test dataset.

About Dataset

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according being ham (legitimate) or spam.

The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

This corpus has been collected from free or free for research sources at the Internet:

-> A collection of 5573 rows SMS spam messages was manually extracted from the Grumble text Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.

About Dataset

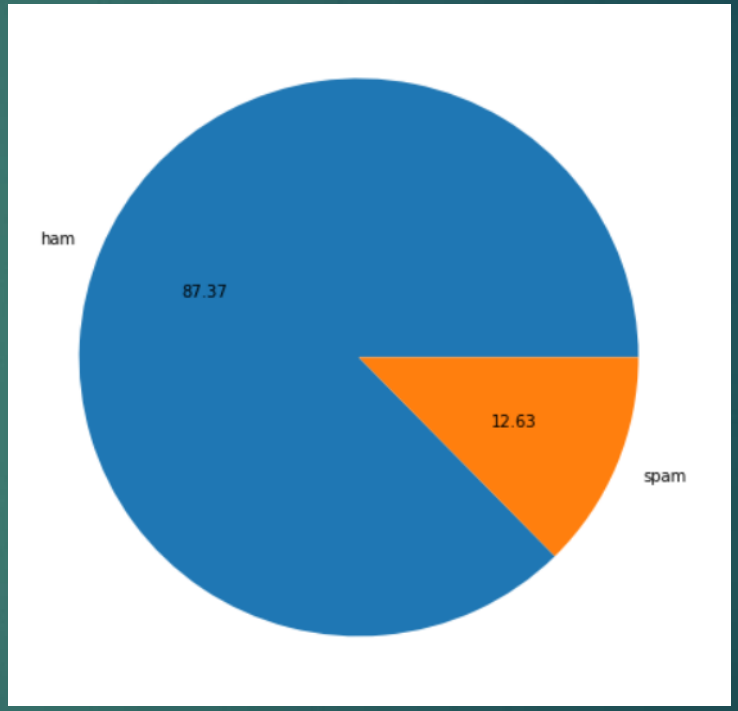
As mentioned above, the column v1 is denoting whether the email is spam or ham, we renamed this column as ‘class_label’ and the column v2 contains the text message of email we renamed this as ‘message’.

There are 3 more unnamed columns having more than 90% NaN values. Hence we dropped these features.

Also, we checked for duplicates and found that there are 403 duplicate entries. We used drop duplicate method to remove these duplicates.

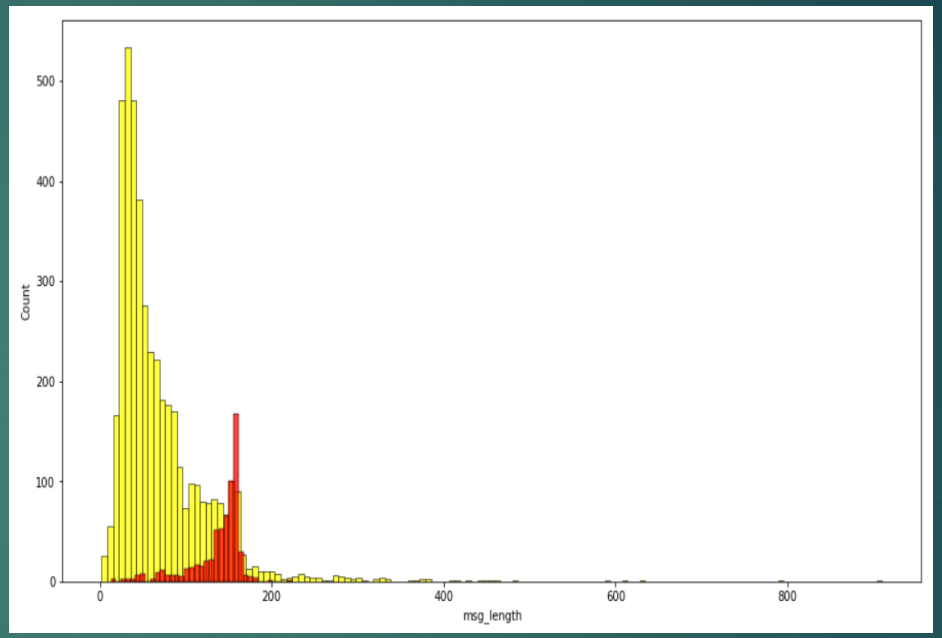
Labelwise distribution of emails

- There are total 5572 rows out of which 4516 mails are labelled as ham and 653 mails are labelled as spam.
- If we look percentagewise, then 87.37% emails are labelled as ham and only 12.63% emails are labelled as spam.



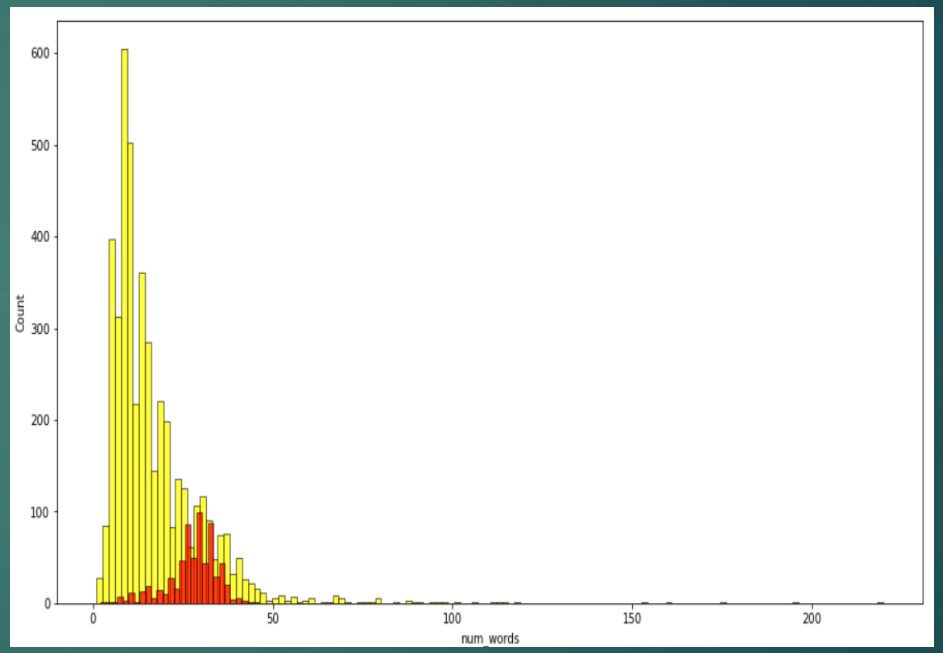
Distribution of message length

- The plot shows the distribution of message length for spam and ham emails.
- The red lines represents message length of spam messages and yellow lines represents message length of ham emails.
- We can say that the message length is maximum for ham emails and minimum for spam emails.



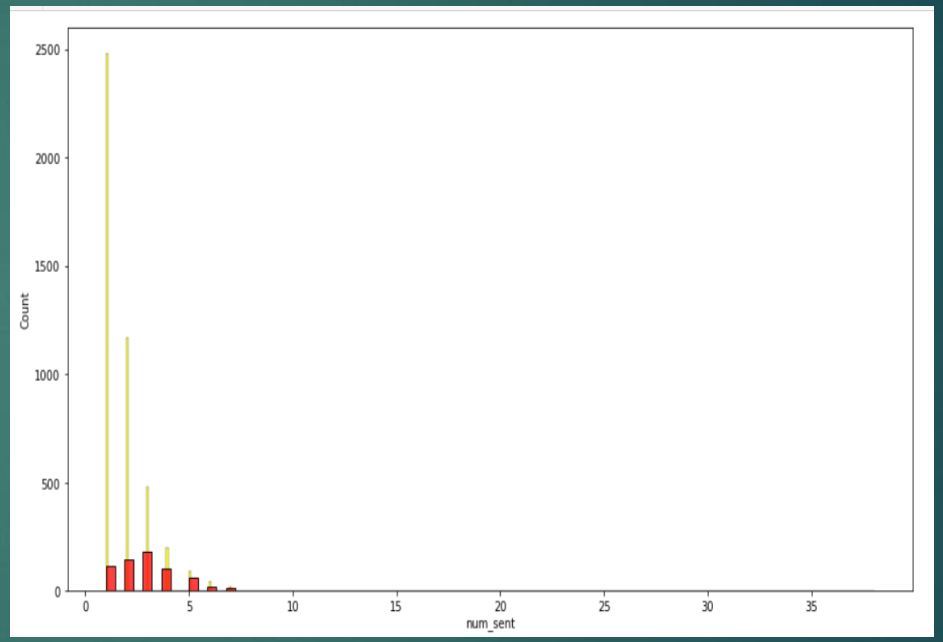
Distribution of Number of words

- The plot shows the distribution of number of words for spam and ham emails.
- The red lines represents number of words of spam messages and yellow lines represents number of words of ham emails.
- We can say that the number of words are maximum for ham emails and minimum for spam emails.



Distribution of Number of Statements

- The plot shows the distribution of number of sentences in ham and spam emails represented by yellow and red respectively.
- The plot shows that the number of sentences are more for ham emails and less for spam emails.



Data Pre-processing

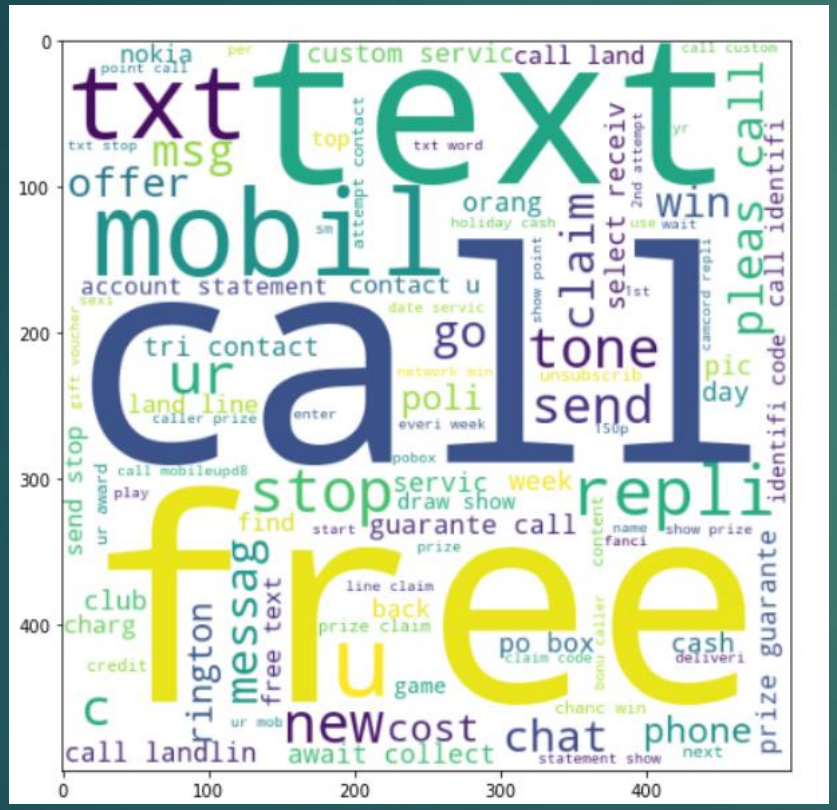
Below are the required steps in data pre-processing:

- Convert the text to lowercase
- Remove the punctuations, digits and special characters
- Tokenize the text, filter out the adjectives used in the review and create a new column in data frame
- Remove the stop words
- Stemming and Lemmatising
- Applying Text Vectorization to convert text into numeric

Word Cloud for Getting Word Sense

- Word Cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.
- The more commonly the term appears within the text being analysed, the larger the word appears in the image generated.
- The enlarged texts are the greatest number of words used there and small texts are the smaller number of words used.

Word Cloud for words in spam emails



From the wordcloud of spam emails it is clear that the words 'call','free','new','mobil','text','txt','offer','repli','claim','stop' etc. are the most common words in spam class.

Word Cloud for words in ham emails

From the wordcloud of ham emails it is clear that the words 'u', 'ur', 'come', 'go', 'got', 'want', 'day', 'time', 'need', 'know' etc. are most common words in ham messages.

Libraries Imported

- Visualization & Data Wrangling Library used

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # ploting
import matplotlib.ticker as ticker
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

- Text Mining

```
#Importing Required libraries
import nltk
import re
import string
from nltk.corpus import stopwords
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

- Machine Learning Model Building Library used

```
#Importing Machine learning Model library
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.svm import SVC, LinearSVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import hamming_loss, log_loss
```

```
import timeit, sys
import tqdm.notebook as tqdm
```

Algorithms Used for Model Building

We used Multinomial Naive-Bayes algorithm to build a spam classifier model.

We also used some other algorithms for comparison and getting better accuracy and precision scores which are as below:

- LogisticRegression
- KNeighborsClassifier
- DecisionTreeClassifier
- RandomForestClassifier
- AdaBoostClassifier
- BaggingClassifier
- GradientBoostingClassifier
- SVC
- XGBClassifier

Result

Sr no.	Algorithm	Accuracy score	Precision score	CV Score
1	KNN	0.894818	1.000000	1.000000
2	MNB	0.948956	1.000000	1.000000
3	RF	0.962877	1.000000	0.989473
4	SVC	0.965197	0.953020	0.963835
5	XGB	0.962104	0.939597	0.957739
6	LR	0.945862	0.936508	0.930299
7	GDB	0.953596	0.928571	0.947339
8	ADB	0.955143	0.917808	0.920298
9	BG	0.955143	0.858824	0.863729
10	DT	0.928074	0.822222	0.817915

Result

- Now we can see that there are 3 algorithms giving best precision score as 1 with good accuracy scores.

- Out of them we will select the Multinomial NB as our final algorithm which gives us precision score 1 and accuracy score 0.948956 with CV score 1.

Conclusion

In the study, we analysed machine learning techniques and their application to the field of spam filtering. A review of the algorithms been applied for classification of messages as either spam or ham is provided. The system architecture of email spam filter and the processes involved in filtering spam emails were looked into. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness and efficiency of any spam filter. The challenges of the machine learning algorithms in efficiently handling the menace of spam were pointed out and comparative studies of the machine learning techniques available in literature was done.

Scope for this Project

i. Modified existing machine learning algorithm.

ii. Make use and classify of a data set including data preparation, classification and visualization.

iii. Score of data to determine the accuracy of spam detection

iv. It provides sensitivity to the client and adapt well to the future spam techniques.

v. It considers a complete message instead of single word with respect to its organization.

vi. It increases security and control.

vii. It reduces IT administration cost and Network resource costs.

Limitations

The limitation of this project are:

i. This project can only detect and calculate the accuracy of spam messages only.

ii. It focus on filtering, analysing and classifying the messages.

iii. Do not block the messages.

Thank You